



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/633,375	07/21/2003	Michael V. Sliger	13768.783.116	5139
47973	7590	12/23/2008	EXAMINER	
WORKMAN NYDEGGER/MICROSOFT			WEI, ZHENG	
1000 EAGLE GATE TOWER				
60 EAST SOUTH TEMPLE			ART UNIT	PAPER NUMBER
SALT LAKE CITY, UT 84111			2192	
			MAIL DATE	DELIVERY MODE
			12/23/2008	PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>	
	10/633,375	SLIGER ET AL.	
	<b>Examiner</b>	<b>Art Unit</b>	
	ZHENG WEI	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) Responsive to communication(s) filed on 19 September 2008.
- 2a) This action is **FINAL**.                  2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) Claim(s) 1-6,8,9 and 11-23 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) Claim(s) \_\_\_\_\_ is/are allowed.
- 6) Claim(s) 1-6, 8-9 and 11-23 is/are rejected.
- 7) Claim(s) \_\_\_\_\_ is/are objected to.
- 8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on \_\_\_\_\_ is/are: a) accepted or b) objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) All    b) Some \* c) None of:  
 1. Certified copies of the priority documents have been received.  
 2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |  |   |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892)                     | 4) <input type="checkbox"/> Interview Summary (PTO-413)           |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ .                                    |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)          | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____.   | 6) <input type="checkbox"/> Other: _____ .                        |

## **DETAILED ACTION**

### ***Remarks***

1. This office action is in response to the amendment filed on 09/19/2008.
2. Claims 24-33 have been canceled.
3. Claims 1, 8, 13 and 17 have been amended.
4. The 35 U.S.C. 101 rejection to claims 24-25 and 29-33 is withdrawn in view of the Applicants cancellation.
5. Claims 1-6, 8-9 and 11-23 remain pending and have been examined.

### ***Response to Arguments***

6. Applicant's arguments filed on 09/19/2008, in particular on pages 10-12, have been fully considered based on the interpretation of the amended claims and the Applicants' specification:
  - The present application according to the specification discloses a method for intra-package delta compression including receiving a plurality of source files (N files); generating plurality of lists (N) of prospective delta inputs (entries), wherein each one of the source files has a corresponding generated list. Each of the lists has (N-1) entries which are used to represent all the received source files except the source file itself for the purpose of as inputs for generating deltas. Therefore, the total number of entries of all the lists are N\*(N-1); selecting one source file as a base file each time for all the source

files, generating (N-1) deltas from the base file and every source file in the list as synthesized file as inputs and thus for the number of N lists, there are total number of N(N-1) generated deltas; Based on the generated deltas and the received source files, generating a directed graph by using each source files as vertex, each deltas as edge, size of delta as weight and additional NULL vertax and size of simple compressed the source file as weight for root as the starting point for further selecting; identifying and selecting the source files in the directed graph by applying minimum spanning tree technology to prevent loop for selecting in the graph and to get the minimize the total size of the base file and deltas; using the linked list to represent the identified source files and other source files that are not covered by the directed minimum spanning tree; generating manifest file and directives file based on the linked list for the final package.

- According to the Examiner's interpretation above, the claim language of amended claim 1 is not clear. Claim 1 recited “an iterator creating a delta for each prospective delta on each files' list of prospective delta inputs”, “generating a delta from the base file and a source file” and “packaging the manifest file, base file and the delta into a self-contained package” [emphasis added]. If the “creating a delta for each prospective delta” is considered as creating a plurality of deltas, it is not clear to the examiner what the “the delta” refers to, a delta or deltas? Moreover, the claim recites a term “creating a linked list” without disclosing what the linked list is, what's the ordering of the

linked list and how does it relate to generate the manifest file? Furthermore, claim 8 recites the steps of constructing a directed graph without providing any relationship between directed graph and generating self-contained package as recited in claim 1. Therefore, the ordinary skill in the art is hard to follow based on the claimed steps to generate the self-contained package.

- At page 11, second paragraph, the Applicants submit that the prior art fails to teach or suggest for each of a plurality of source files , generating a list prospective delta inputs, the prospective delta inputs including an entry for each other unique source file in the plurality of source files. It should be noted that based on the Examiner's interpretation above, the prospective delta inputs is merely a list of file names which indicates the files that are used to generate deltas with the base file. As Zan disclosed at ABSTRACT by performing a sequence of pairwise delta compressions for all the collection of files, it is clear that to perform pairwise delta compressions for all the files, it has to have a list of files that directs the specific delta compressor to find an optimal solution.
- At page 11, second paragraph, the Applicants also submit that the prior art also fails to teach or suggest an iterator creating a delta for each prospective delta on each file's list of prospective delta inputs. It should be noted that, as Zan disclosed above for pairwise compression for all the collection files, it has to know which two file (pairwise) to perform delta compression, after each pairwise compression, a delta has to be generated for each of all pairwise compressions.

- At page 11, second paragraph, the Applicants also submit that the prior art also fails to teach or suggest processing the source files into a base file for a package based upon a minimal package size. Same as address above, because of Zan's pairwise compression, each file in the collection has to be compressed with other files in the collection. During this process, the file is treated as base file and the other files are as synthesized files accordingly to complete pairwise compressions. Moreover, Zan discloses finding an optimal delta encoding for a collection of file for a minimal package size (see for example, p.3, section Delta compression based on optimum branching and Fig.1).
- At page 11, second paragraph, the Applicants also submit that the prior art also fails to teach or suggest generating a manifest file by creating a linked list of the plurality of source file, the manifest file comprising instructions needed to perform an extraction , the instructions being particularly ordered in the manifest file and the ordering of the instructions corresponding to an ordering of the linked list, and the manifest file comprising a delta section, a copy section, a verify section and a delete section. It should be noted, claim language merely discloses the manifest file comprising four sections: a delta section, a copy section, a verify section, and delete section. But does not explicitly disclose what these sections are and what their functions. Therefore, as Draper disclosed "delta lookup table", "file of modification data blocks", and "basis index table", it also contains a lot of section in the files.

- At pages 11-12, the Applicants submit that the prior art fails to teach or suggest all the limitation of the claims as now recited. However, as addressed above, claim language merely discloses the manifest file comprising four sections: a delta section, a copy section, a verify section, and delete section. But does not explicitly disclose what these sections are and what their functions. Therefore, as Draper disclosed "delta lookup table", "file of modification data blocks", and "basis index table", it also contains a lot of section in the files.

***Claim Rejections - 35 USC § 112***

7. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.
8. Claims 1-6, 8-9 and 11-12 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim1:

Claim 1 recited "an iterator creating a delta for each prospective delta on each files' list of prospective delta inputs", "generating a delta from the base file and a source file" and "packaging the manifest file, base file and the delta into a self-contained package" [emphasis added]. If the "creating a delta for each prospective delta" is considered as creating a plurality of deltas, it is not clear to the examiner what the "the delta" refers to, a delta or deltas? Moreover, the claim

recites a term “creating a linked list” without disclosing how to creating the linked list, what’s the ordering of the linked list and how does it relate to generate the manifest file? For the purpose of compact prosecution, the Examiner treats “the delta” as --the deltas—

Claim 2-6, 8-9 and 11-12:

Claims 2-6, 8-9 and 11-12 depend on claim 1 and thus they are also rejected for the same reason.

Claim 8:

Claim 8 recites the steps of constructing a directed graph without providing any relationship between directed graph and generating self-contained package as recited in claim 1. Therefore, it is not clear to the Examiner where the directed graph is used in the generating the package. The examiner treats the directed graph for creating the linked list as in amended claim 1.

Claim 9:

Claim 9 depends on claim 8, therefore, it is also rejected for the same reason.

***Claim Rejections - 35 USC § 103***

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

10. Claims 1, 4-6, 8, 11, 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Zan (Zan et al, “Cluster-Based Delta Compression of a Collection of Files”) in view of Draper (US 6,604,236) in further view of Crudel (US 2002/0099726)

Claim 1:

Zan discloses a method in a computing environment for intra-package delta compression, the method comprising:

- receiving information corresponding to a plurality of source files (see for example, p.2, left column, lines 10-13, “obtaining optimal compression of a collection of n files”);
- for each of the plurality of source files, generating a list of prospective delta inputs, the prospective delta inputs including an entry for each unique source file in the plurality of source files (see for example, p.3, section 2.1, “Given a collection of n files we construct a complete directed graph...where each node corresponds to a file...”; also see ABSTRACT, pairwise compressions);

Art Unit: 2192

- an iterator creating a delta for each prospective delta on each file's list of prospective delta inputs (see for example, ABSTRACT, pairwise compressions; also see, p.2, section Delta compression based on optimum branching and Figure 1)
- processing the source files into a base file for a package based upon package size (file/node weight) (see for example, p.3 Fig.1, example of a directed and weighted complete graph; also see p.3, left column, last paragraph to right column first paragraph about generating base file (file 1));
- generating a delta from the base file and source file (see for example, p.3, right column, first paragraph, “file2 and 3 are compressed by computing a delta with respect to file 1”);

But Zan does not explicitly disclose cited limitations about calculating signature, saving file name, signatures and instructions in manifest file and packaging the base file and the delta into a self-contained package.

However, Draper in the same analogous art of delta compression discloses

- calculating signatures for each of the plurality of source files (see for example, Fig.3A step 125, “Generate checksum for file” and related text);
- generating instructions needed to perform an extraction (see for example, col.1, lines 52-60, “basis index table”, “file of modification data blocks” and “delta lookup table” and related text)

Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to combine Zan and Draper to generate compressed files including signature and instruction. One would have been motivated to do so to update the content of a copy of the original file system without having to generate a copy of every file and data block for the new content of the original file system as suggested by Draper (see for example, ABSTRACT, last paragraph)

Zan discloses a linked list of the plurality of source files (optimal branching) (see for example, p.3, Figure 1. Example of a directed and weighted complete graph, a linked list that covers edges (0,1), (1,2), (1,3) and (3,4)), but does not explicitly discloses generating a manifest file by creating the linked list. However, as Draper also disclosed above for generating manifest file (basis index table, delta lookup table...), it is also obvious that the manifest file including different section (basis index table, delta lookup table...) has to be generated by Zan's method including using the linked list as address above.

But neither Zan nor Draper discloses saving the instructions, file name (entry name) and signatures in a manifest and further packaging the manifest file, base file and delta into a self-contained package. However, Crudele in the same analogous art of distribution of file updates, discloses creating a software package (see for example, paragraph [0013], [0025] "distribution package file) with including a distribution package file, the delta distribution package file comprising a delta file (see for example, paragraph [0025]) and signatures

(crc32) (see for example, paragraph [0025]). Crudele further discloses that this package is created by administrator via graphic user interface (GUI). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use Crudele's method and its GUI to also include the base file in the package for distribution. It is also obvious to save all the information about instruction, signature and file name in a one manifest file instead of separate file in the software package.

Claim 4:

Zan discloses the method of claim 1 wherein the first source file and the second source file are not different versions of the same file. (see for example, Fig.1, and related text, also see, p.3, left column, lines 8-15, "collection of files")

Claim 5:

Zan discloses the method of claim 1 wherein the first source file and the second source file are not different language translations of the same file. (see for example, Fig.1, and related text, also see, p.3, left column, lines 8-15, "collection of files")

Claim 6:

Zan discloses the method of claim 1 wherein the first source file and the second source file are different language translations of the same file. (see for example,

Fig.1, and related text, also see, p.3, left column, lines 8-15, "collection of files")

Claim 8:

Zan also discloses the method of claim 1 further comprising constructing a directed graph by adding vertex (node) and edges, giving weight to each edge, adding null vertex and related edges and weight , and selecting the first source file based on information in the directed graph. (see for example, Fig.1, crating the directed graph and related text, "edge" and "node")

Claim 11:

Zan also discloses the method of claim 1 further comprising, providing the package to a recipient, the recipient applying the delta to the first source file to synthesize the second source file (see for example, p.1, right column, lines 1-8, "sender and receiver both possess a reference file").

Claim 12:

Claim 12 is computer product version for performing the claimed method as in claim 1 addressed above, wherein all claimed limitation functions have been addressed and/or set forth above and certainly it is well known in the computer art to run and/or practice as such computer product which has its computer-executable instructions stored on a computer-readable storage medium so that this computer product would be executed by the computer system to perform the

method addressed in claim 1 above to realize its functionality. Therefore, they also would have been obvious by Zan, Draper and Crudele.

11. Claim 2 is rejected under 35 U.S.C. 103(a) as being unpatentable over by Zan, Draper and Crudele in view of Forbes (Forbes et al., US 6,381,742 B2).

Claim 2:

Zan, Draper and Crudele disclose the method as in claim 1 above, but do not disclose the method further comprising, packaging data for directing a client extractor to synthesize a target file corresponding to the second source file from the base file and the delta. However, Forbes in the same analogous art of software package management discloses a manifest file (package data) to manage the installation, execution. (see for example, Fig.2A, element 207 and related text. Also see abstract about the manifest file). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to include the manifest file into the software package to provide configuration information to the software installation. One has motivation to do so to automatically install software package without requiring manual intervention by the user. (see for example, col.3, lines 30-34, “Because the manifest file contains the location of the distribution units for any dependencies, the software package manager can acquire and install the dependencies without requiring manual intervention by the user.”)

12. Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over by Zan, Draper and Crudele in view of Henry (Craig James Henry, US 6,131,192).

Claim 3:

Zan, Draper and Crudele disclose the method as in claim 1 above, but does not disclose the method further comprising, setting at least one file name by which a client extractor may synthesize a target file corresponding to the second source file from the base file and the delta. However, Henry in the same analogous art of software installation discloses a method for setting up the software product name. (see for example, Fig.4B, steps 415-445 and related text). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to set the target file name for decompressed file. One has motivation to do so to identify file to decompress and set right file name to further installation and execution. (see for example, col.18, lines 26-32)

13. Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Zan in view of Weiss (Mark Allen Weiss, "Data Structures & Algorithm Analysis in C++").

Claim 9:

Zan discloses the method as in claim 8 above wherein a branching B of a directed graph G does not contain a cycle, but does not disclose using minimum spanning tree or like algorithm to the directed graph to eliminate loop. However, Weiss in the same analogous art of eliminate loop in graph discloses a method of using minimum spanning tree. (see for example, p.356-362, "Prim's Algorithm")

and “Kruskal’s Algorithm”). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use minimum spanning tree algorithm to eliminate loop in Zan’s directed graph. One has motivation to do so to prevent loop in Zan’s directed graph as once required by Zan (see for example, p.3, left column, line 19, “B does not contain a cycle”).

14. Claims 13, 14, 15 and 21-23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Zan (Zan et al, “Cluster-Based Delta Compression of a Collection of Files”) in view of Draper (US 6,604,236 236) and in view of Sliger (Sliger et al., US 6,216,175)

Claim 13:

Zan discloses in a computing environment, a method to compress a collection of files to generate a plurality of deltas and base files in a package, (see for example, p.1, right column, lines 23-26, “using delta compression to better compress large collections of file where it is not obvious at all how to efficiently identify appropriate reference and target files”). Zan further discloses a linked list of the plurality of source files (optimal branching) (see for example, p.3, Figure 1. Example of a directed and weighted complete graph, a linked list that covers edges (0,1), (1,2), (1,3) and (3,4)), but does not explicitly discloses generating a manifest file by creating the linked list.

However, Draper in the same analogous art of delta compression discloses generating instructions needed to perform an extraction (manifest file) (see for

example, col.1, lines 52-60, “basis index table”, “file of modification data blocks” and “delta lookup table” and related text). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to combine Zan and Draper to generate compressed files with manifest file including extracting instruction which has multiple sections. One would have been motivated to do so to update the content of a copy of the original file system without having to generate a copy of every file and data block for the new content of the original file system as suggested by Draper (see for example, ABSTRACT, last paragraph). But they do not explicitly disclose how to decompress them. However, Sliger in the same analogous art of software updating and patching discloses a method comprising:

- receiving a package(see for example, Fig.3, item 54 and related text, “Patch File”); and
- synthesizing a target file by applying a delta in the package to a base file to synthesize a target file (see for example, Fig.3, items 54, 58 and related text, also see Fig.7, “user’s computer” and related text)

Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use Sliger’s decompressing method to decompress and generate target files from Zan’s compressed file. One has motivation to do so in order to reduce communication or storage costs as once pointed by Zan (see for example, p.1, left column, section 1, Introduction)

Claim 14:

Zan, Draper and Sliger disclose the method as in claim 13 above, Zan further discloses the method wherein applying the delta to the base file comprises applying the delta to a base file included in the package (see for example, p.2, left column, lines 41-44, “compressing and uncompressing an entire collection”, also see Fig.1 and related text).

Claim 15:

Zan, Draper and Sliger disclose the method as in claim 13 above. Zan also discloses the method wherein applying the delta to the base file comprises applying the delta to a base file synthesized from another delta and another base file (see for example, p.3, right column, lines 1-5, “sequence for compression”)

Claim 21:

Zan, Draper and Sliger disclose the method as in claim 13 above. Zan also discloses the method further comprising, applying another delta to the synthesized target file to synthesize another target file (see for example, fig.1 and related text, also see lines 1-5, “files 1...4”).

Claim 22:

Zan, Draper and Sliger disclose the method as in claim 13 above. Zan also discloses the method further comprising, applying at least two deltas to a

common base file to synthesize at least two target files (see for example, fig.1 and related text, also see lines 1-5, "The optimal sequence for compression is (0,1), (1,2), (1,3)").

Claim 23:

Zan, Draper and Sliger disclose the method as discussed in claim 13 above. It is well known in the computer art that said method can be practiced and stored in the computer-readable medium. Therefore, this claim is also obvious by Zan and Sliger.

15. Claims 16 and 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over Zan (Zan et al, "Cluster-Based Delta Compression of a Collection of Files") in view of Draper and Sliger and in further view of Forbes (Forbes et al., US 6,381,742 B2).

Claim 16:

Zan, Draper and Sliger disclose the method as in claim 13 above, but do not disclose using the data file to determine to which base file each delta is to be applied. However, Forbes in the same analogous art of software package management discloses a manifest file (package data) to manage the installation, execution. (see for example, Fig.2A, element 207 and related text. Also see abstract about the manifest file). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to include the

manifest file into Zan and Sliger's method to provide configuration information to the software installation. One has motivation to do so to automatically install software package without requiring manual intervention by the user as once suggested by Forbes. (see for example, col.3, lines 30-34, "Because the manifest file contains the location of the distribution units for any dependencies, the software package manager can acquire and install the dependencies without requiring manual intervention by the user.")

Claim 17:

Zan, Draper and Sliger disclose the method as in claim 14 above, but do not disclose the method wherein the manifest file comprises a set of instructions including instructions that identify a particular base file to which a particular delta file is to be applied. However, Forbes in the same analogous art of software package management discloses a manifest file (data file) to manage the installation, execution. (see for example, Fig.2A, element 207 and related text. Also see abstract about the manifest file). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to include the manifest file into Zan and Sliger's method to provide configuration information to the software installation. One has motivation to do so to automatically install software package without requiring manual intervention by the user as once suggested by Forbes (see for example, col.3, lines 30-34, "Because the manifest file contains the location of the distribution units for any

dependencies, the software package manager can acquire and install the dependencies without requiring manual intervention by the user.”)

16. Claims 18-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Zan (Zan et al, “Cluster-Based Delta Compression of a Collection of Files”) in view of Sliger and Draper and in further view of Henry (Craig James Henry, US 6,131,192).

Claim 18:

Zan, Draper and Sliger disclose the method as in claim 13 above, but do not disclose the method further comprising, executing a setup program. However, Henry in the same analogous art of software installation discloses the method comprising setting up the software product. (see for example, Fig.3, item 130, 135 and 140 and related text, “Place decompressed file in the temporary storage space”, “Begin setting up the software product”). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to further execute the setup program to install files which are decompressed by Zan and Sliger. One has been motivated to do so to simplify and streamline the process of installing a software product on a computer as once suggested by Henry (see for example, col.1, lines 48-50)

Claim 19:

Zan, Draper, Sliger and Henry disclose the method as in claim 18 above, Henry

further discloses the method wherein the setup program is executed after each delta has been applied to a corresponding base file. (see for example, Fig.3, item 120, 125, 130, 135 and related text, "Place decompressed file in the temporary storage space"). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to further execute the setup program to store files in the temporary directory which are decompressed by Zan and Sliger and check the decompression status. One having been motivated to do so to simplify and streamline the process of installing a software product on a computer as once suggested by Henry (see for example, col.1, lines 48-50)

Claim 20:

Zan, Draper and Sliger disclose the method as in claim 13 above, but do not disclose the method further comprising, deleting the deltas from a temporary directory. However, Henry in the same analogous art of software installation discloses the step to delete files form temporary storage space. (see for example, Fig.3, step 155, "Delete Files From temporary storage space" and related text). Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to delete the deltas files in the temporary directory. One having been motivated to do so to reduce storage costs as once suggest by Zan (see for example, p.1, left column, section 1, Introduction)

***Conclusion***

17. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Applicant's arguments with respect to claims rejection have been considered but are not persuasive. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

18. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Zheng Wei whose telephone number is (571) 270-1059 and Fax number is (571) 270-2059. The examiner can normally be reached on Monday-Thursday 8:00-15:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The

fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Any inquiry of a general nature of relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571- 272-1000.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Z. W./  
Examiner, Art Unit 2192

/Tuan Q. Dam/  
Supervisory Patent Examiner, Art Unit 2192